DECUS
PROGRAM LIBRARY

DECUS NO.	8-397
TITLE	8K-EDITOR
AUTHOR	Bill Donelson
COMPANY	The Choate School Wallingford, Connecticut
DATE	March 15, 1971
SOURCE LANGUAGE	PAL-D

8K-EDITOR

DECUS Program Library Write-up — DECUS NO. 8-397

NOTE: I wish to thank Roger Collins, without whose complaints, an editor of this calibre would not have been possible.

PART ONE - Input/Output Specifications

I. Disk Systems

A. My editor uses the "command decoder" to set up I/O. This must be set as internal block #5 in the SAM directory. This "Command Decoder" is set up by version AF of the disk monitor system.

B. The possible file types are:

T:	Low Speed paper
R:	H/S paper tape
S: filename	disk dataset
↵ (carriage return)	null device

NOTE: These may be mixed in any order for the input and output prompts. I/O filenames may be the same.

A typical ".CD." - user "conversation" might be: (underlined portions are typed by the machine)

. Edit ↵	
*OUT - S:BILL ↵	Output to disk (BILL)
*	
*IN - S:BILL ↵	Input from disk (BILL)
*	
*OPT - ↵	None specified
E=>	Editor Command Level

Notice that disk I/O files may be the same. Only one file may be specified for each I/O prompt.

The option prompt ignores all but a "D" answer. A "D" answer specifies deletion of the input file from disk after it has been read. An output file will not be cleared until output from the editor is attempted.

NOTE - The high speed (H/S) reader and punch are always enabled.

II. Non-disk Systems

A. Non-disk Systems must have a H/S paper tape system. The starting address is at 01000_8 rather than 03200_8 as in the disk system.

B. All I/O commands use the H/S equipment. This system will act in exactly the same way as if the disk system user had specified the following initialization:

*OUT - R: ↓
*
*IN - R: ↓
*
*OPT - ↓

CAUTION: If an "E" command is given, control will be returned to location 7600_8 - field Ø as if a normal return to the disk monitor were expected.

PART TWO - Commands and Formats

I. Generalized Command Format

The generalized format is:

E=> k nnn,mmm dString d bstring ↓
ⓐ ⓑ ⓒ ⓓ ⓔ ⓕ ⓖ ⓗ ⓘ

where:

ⓐ is Command Level Prompt from Editor

ⓑ The one letter command

ⓒ "nnn" is a line or column # (optional)

ⓓ ",mmm" is a second line # (optional)

ⓔ is the search string delimiter (originally set to $") (optional)

ⓕ-ⓗ are search strings of one or more characters separated by a delimiter

ⓘ is a return (carriage return char.)

Parts ⓒ thru ⓘ are optional depending on the command.

II. Special Characters

1). control-L (214_8) deletes the entire command and returns to the editor command level.

2). rubout - (377_8) erases the immediately preceding character which is not itself a rubout. If the command character is erased, a return to command level will follow.

3). Space (240_8) are ignored except in search strings.

4). Slash ("/") the last line # is used in its place (except in search strings).

5). Dot or period (".") same as slash only use current line #.

6). "+", "-" plus, minus, used to add or subtract line #'s. "1-3" or ".+1"

III. Command List

COMMAND	EXPLANATION
A	append
B	bottom
C	change
D	delete
E	end
F	fix the tab
G	get
H	H/S read
I	insert
L	list
N	next
O	output
P	H/S punch
Q	quit
R	read
S	search
T	top
U	unset verification
V	set verification
X	unset line #'s
>	next line
<	previous line
.	print current line #
/	print last line #
#	set line #'s
↑	change base
:	print current base
$	change the delimiter
"	repeat the previous command

PART THREE - Commands and Explanations

Command

(A) append

Form: E=> A ↓

Append to the end of the current buffer. Input format the same as on the "I"nsert command. Executes the same as E=> 1/+1 ↓

(B) bottom

(a) Form: B ↓

Print last line and set current line pointer to last line.

(b) Form: B n ∂ String ↓

Find the last occurrence of "String" in the buffer and print that line. If "n" is omitted assume 1 and start search with that line

(C) change

(a) Form: C n,m ↓

Executes the same as :

E=> Dn,m ↓
E=> In ↓

(b) Form: C n,m ∂ fstring ∂ gstring ↓

Change "fstring" to "gstring" in lines "n" thru "m" . Only the first occurrence on each line is changed. If "n,m" is omitted, "." is assumed.

(D) delete

Form: Dn,m

delete lines "n" thru "m" from the current buffer.

(E) end

Form: E ↓

Perform "Nexts" until no input is left and then close the output file. A return to the DISK MONITOR ($76\emptyset\emptyset_8$) follows.

(See notes on Control-C at end of tutorial)

(F) fix the logical tab

Form: F n ↓

Set the tab at "n" columns and each multiple of "n" following it. A control-I (tab) is echoed as the appropriate # of spaces.

(G) get

(a) Form: G n ↵

Get and print the next line which begins with a character other than tab (control-I, 211_8) or space (240_8). Start search with line "n". If "n" is omitted assume ".+1".

(b) Form: G n ∂ String ↵

Starting with line "n", find the first line which contains "String" in the first "tab"* columns. If "n" is omitted, ".+1" is assumed

* (Logical tab is initially set at 8 columns.)

(H) high speed paper tape READ

Form: H ↵

Same as READ command, except always use H/S reader.

(I) insert

Form: I n ↵

Insert, before line "n", as many lines as are entered from the keyboard. Each line is terminated by a " ↵ ", line-feeds are ignored. Insert mode is prompted by "nnn>"; where "nnn" is the line to be entered. NOTE - if the "X" command has been used, ">" will be the only prompt. Rubouts may be used to erase incorrect characters on the current line. Insert mode is exited by a control-L (214_8) character; the current line (before a control-L) is ignored. If "n" is omitted, "1" is assumed.

(L) list

Form: L n,m ↵

List lines "n" thru "m", inclusive. If "m" is omitted, list "n" only. If "n" is omitted or is equal to Ø, list the entire program. If "X" has been used, print only the lines themselves, if "X" has not been set, number the lines ("nnn >").

(N) next buffer

Form: N n ↵

Output the current buffer to the output file, delete it from core, and then read in a new buffer from the input file. (See Read command and also the notes on Special Characters at the end of the tutorial.)

(O) output

Form: O n,m ↵

Output to the output file, lines "n" thru "m". (No line numbers are in the output.) If "m" is omitted, only "n" is outputted. If "n" = Ø, the entire buffer is outputted. (If "n" omitted it is assumed to be Ø.) (See notes on control-C and Special Characters.)

(P) punch

Form: P n,m ↵

Same as "O", except punch on H/S punch

(Q) QUIT

Form: QUIT ↵

Return to disk monitor. If a disk output file has been specified, then update all disk files and return to the monitor. If a non-disk output file has been specified, NO disk files will be written or destroyed; the disk will remain as it was before the editor was called.

EXAMPLES:

```
.EDIT ↵
*OUT - S:TNUM ↵
*
*IN - R: ↵
*
*OPT - ↵
E =>

    various commands
    issued here.

E => QUIT ↵
.
```

Note - all updates are performed. The file "TNUM" will have been created on disk.

```
.EDIT ↵
*OUT - R:
*
*IN - S:TEST
*
*OPT - D ↵
E =>

    various commands
    given.

E => QUIT ↵
```

Note - Nothing will be changed on disk, REGARDLESS of the "D" option.

(R) read from input file

Form: R ↵

Read from the input file until:

(a) The buffer exceeds 3/4 capacity. (No characters are lost -- the read terminates after the current line is read.) This allows room for work - (i.e. Append, Insert, Change, etc.).

ⓑ A control-L (214_8) is encountered in the input string. The read may be continued by issuing another "R" command. The same rules (ⓐ + ⓑ) apply.

Ⓢ search

Form: S n ↵ String ↵

Search, starting with line "n", for the next line containing "String". If "n" is omitted assume ".+1".

Ⓣ top

ⓐ Form: T ↵

Same as: L 1 ↵

ⓑ Form: T ↵ String ↵

Same as S1 ↵ String ↵

Ⓤ unset verification of Change Command

Form: U (note - no " ↵ " needed)

Do not print the corrected line after a change of the form -

[C n,m ↵ fstring ↵ gstring ↵]

This is useful if a great many lines are to be changed - Example =

E=> C 1, 395AXYZ123456 ↵

where an "A" may occur in every line.

Ⓥ verify the change

Form: V (no " ↵ "

Opposite of "U" command.

Ⓧ unset line numbering from Teletype

Form: X (no " ↵ "

read the title, you nitwit.

\# set line #'s

Form: # (no " ↵ "

⊙ print current line #

Form: E=> . =nnn ↓

where "nnn" is the current line #.

/ print last line #

Form: same as " ⊙ "

↑ change current base to either Octal or Decimal

Form: ↑ (k)

where (k) is the letter: D for deciaml #'s or O for octal #'s

(O is assumed if not a D typed)

EXAMPLE:

E = > ↑ OCTAL ↓ (a "3" was typed after the " ↑ ")
E = > ↑ DECIMAL ↓ (a "D" was typed after the " ↑ ")

NOTE - This affects only Command Strings and line #'s - NOT the actual buffer.

< previous line

Form: < (no " ↓ ")

Same as L . -1 ↓

> Same as < except L . +1

: Print current base

Form: E = >: base ↓

where base is either: OCTAL or DECIMAL

$ change the delimiter ("↵")

Form: E = > $ <

DELIMITER > (k)

where (k) is a one-letter, (printing) character. It is advisable not to use ⊙ + - / or the digits Ø → 9. (obviously)

(") (quote) repeat the previous command.

Form: E = > " (no " ↓ "

See title.

NOTE - will not work on "Change Command" occasionally because of internal structure of editor.

This is useful in:

EXAMPLE

E = > S$ABC ↓

ØØ3> A, ABCD3 ↓

E = > "

ØØ7> AX, AY, AZ, ABC ↓

E = > "

Ø95 > ABCDE ↓

E = > "

? (no "ABC" found in buffer.)

E = > "

? (no "ABC" found in buffer.)

E = >

PART FOUR - Special Characters and Options

1. Control-C (↑ C) ($2Ø3_8$)

A. During printout of a Command. — Will return to Command Level.

B. During a Punch, Output, Next, End — Will return to command level after the current line has been punched (outputted). The current line pointer (".") will be set to the next line in the buffer.

EXAMPLE

E => O 7, .+9 ↓	.=3Ø at this time
?)	(control-C typed
E => . .= 15 ↓	(last line outputted was line 14
E => D 1, 14 ↓	delete lines already punched
E => E ↓	
? ↓	↑ C (control-C typed)
E => D 1, .-1 ↓	delete and continue
E => E	
.	Disk Monitor Level

II. Control-L (214_8)

A. " ↑ L" are put at the end of each buffer or after O, P, N, E commands. A control-L is not put after a control-C half of the punch, etc. However, it will be put at the end of the buffer.

B. Are used to halt the input (H, R) commands. This halt is done rather than the buffer check (See READ command) if the buffer is less than 3/4 full.

C. If disk files are used, the file is closed when two control-L's in succession are found. Two control-L's are placed at the close of an output file.

That's all I can think of —— Please address any questions to:

Bill Donelson
83 Wychewood Road
Memphis, Tennessee 38117

Thanx very much,

Bill Donelson